基于https的tomcat-session集群实验

实验环境：rhel7.3虚拟机3台、关闭防火墙、SELinux、配置好yum

nginx--eth0：192.168.4.1；eth1：192.168.2.1

web1 - eth0:192.168.4.2

web2 - eth0:192.168.4.3

##安装nginx

[root@localhost ~]# useradd nginx

[root@localhost ~]# tar -xf nginx-1.8.0.tar.gz

[root@localhost ~]# yum -y install gcc gcc-c++ pcre pcre-devel openssl-devel

[root@localhost ~]# cd nginx-1.8.0/

[root@localhost nginx-1.8.0]# ./configure --prefix=/usr/local/nginx --user=nginx --group=nginx --with-http\_ssl\_module

[root@localhost nginx-1.8.0]# make && make install

[root@localhost nginx-1.8.0]# cd /usr/local/nginx/conf/

[root@localhost conf]# openssl genrsa -out cert.key

Generating RSA private key, 1024 bit long modulus

..........................++++++

............................................++++++

e is 65537 (0x10001)

[root@localhost conf]# openssl req -new -x509 -key cert.key -out cert.pem

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:CN

State or Province Name (full name) []:beijing

Locality Name (eg, city) [Default City]:bejing

Organization Name (eg, company) [Default Company Ltd]:tarena

Organizational Unit Name (eg, section) []:tarena

Common Name (eg, your name or your server's hostname) []:mak^H

Email Address []:mak@tedu.cn

[root@localhost conf]# ls

cert.key koi-utf scgi\_params

cert.pem koi-win scgi\_params.default

fastcgi.conf mime.types uwsgi\_params

fastcgi.conf.default mime.types.default uwsgi\_params.default

fastcgi\_params nginx.conf win-utf

fastcgi\_params.default nginx.conf.default

#开启加密虚拟主机

[root@localhost conf]# grep -v ^# nginx.conf | grep -v ^$

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

#log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#gzip on;

server {

listen 80;

server\_name localhost;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html;

index index.html index.htm;

}

#error\_page 404 /404.html;

# redirect server error pages to the static page /50x.html

#

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

# proxy the PHP scripts to Apache listening on 127.0.0.1:80

#

#location ~ \.php$ {

# proxy\_pass http://127.0.0.1;

#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000

#

#location ~ \.php$ {

# root html;

# fastcgi\_pass 127.0.0.1:9000;

# fastcgi\_index index.php;

# fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;

# include fastcgi\_params;

#}

# deny access to .htaccess files, if Apache's document root

# concurs with nginx's one

#

#location ~ /\.ht {

# deny all;

#}

}

# another virtual host using mix of IP-, name-, and port-based configuration

#

#server {

# listen 8000;

# listen somename:8080;

# server\_name somename alias another.alias;

# location / {

# root html;

# index index.html index.htm;

# }

#}

# HTTPS server

#

server {

listen 443 ssl;

server\_name localhost;

ssl\_certificate cert.pem;

ssl\_certificate\_key cert.key;

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

location / {

root html;

index index.html index.htm;

}

}

}

[root@localhost conf]# cd ..

[root@localhost nginx]# ./sbin/nginx

[root@localhost nginx]# netstat -antpu | grep nginx

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN 6749/nginx: master

tcp 0 0 0.0.0.0:443 0.0.0.0:\* LISTEN 6749/nginx: master

##启动nginx服务，确认监听80和443端口

##部署tomcat服务器

[root@localhost ~]# tar -xf apache-tomcat-8.0.30.tar.gz

[root@localhost ~]# yum -y install java-1.8.0-openjdk

[root@localhost ~]# mv apache-tomcat-8.0.30 /usr/local/tomcat

[root@localhost ~]# keytool -genkeypair -alias tomcat -keyalg RSA -keystore /usr/local/tomcat/keystore

输入密钥库口令:

再次输入新口令:

您的名字与姓氏是什么?

[Unknown]: li

您的组织单位名称是什么?

[Unknown]: tarna

您的组织名称是什么?

[Unknown]: tarena

您所在的城市或区域名称是什么?

[Unknown]: beijing

您所在的省/市/自治区名称是什么?

[Unknown]: beijing

该单位的双字母国家/地区代码是什么?

[Unknown]: cn

CN=li, OU=tarna, O=tarena, L=beijing, ST=beijing, C=cn是否正确?

[否]: y

输入 <tomcat> 的密钥口令

(如果和密钥库口令相同, 按回车):

再次输入新口令:

[root@localhost ~]# sed -n '85,87p' /usr/local/tomcat/conf/server.xml

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"

maxThreads="150" SSLEnabled="true" scheme="https" secure="true"

keystoreFile="/usr/local/tomcat/keystore" keystorePass="123456" clientAuth="false" sslProtocol="TLS" />

#修改tomcat配置文件，启动加密协议

[root@localhost ~]# /usr/local/tomcat/bin/startup.sh

Using CATALINA\_BASE: /usr/local/tomcat

Using CATALINA\_HOME: /usr/local/tomcat

Using CATALINA\_TMPDIR: /usr/local/tomcat/temp

Using JRE\_HOME: /usr

Using CLASSPATH: /usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar

Tomcat started.

[root@localhost ~]# netstat -antpu | grep java

tcp6 0 0 :::8080 :::\* LISTEN 7141/java

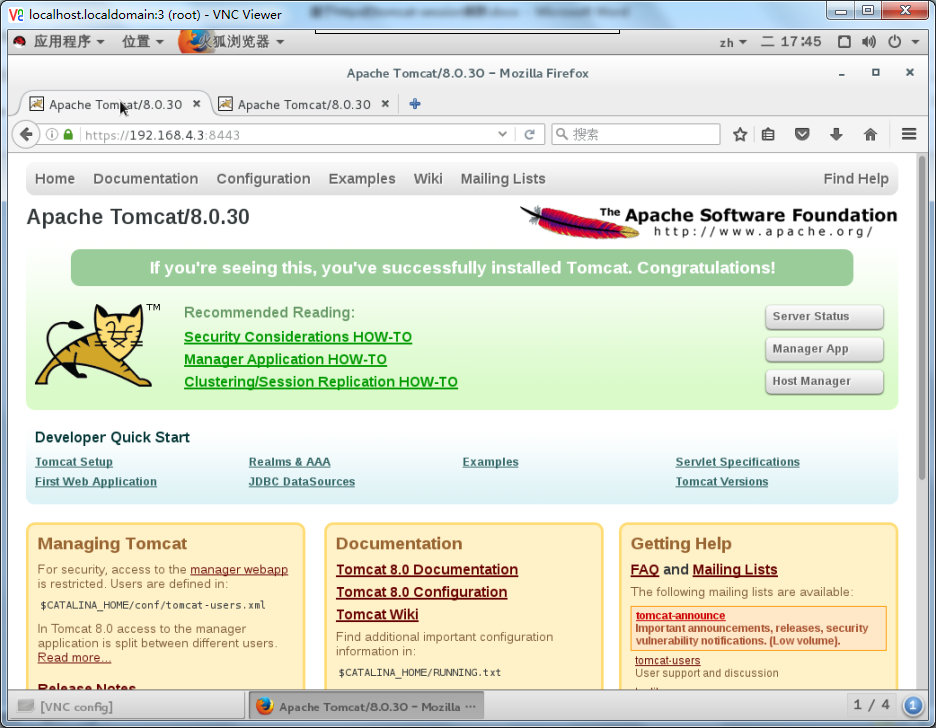
tcp6 0 0 :::8443 :::\* LISTEN 7141/java

tcp6 0 0 :::8009 :::\* LISTEN 7141/java

#启动服务，确认监听端口

#使用firefox测试

#提示不安全的连接，添加例外，得到tomcat的默认页面



[root@localhost ~]# cat /usr/local/tomcat/webapps/ROOT/test.jsp

<html>

<body bgcolor="red">

<center>

<%String s = session.getId();%>

<%=s%>

<h1>tomcatA</h1>

</center>

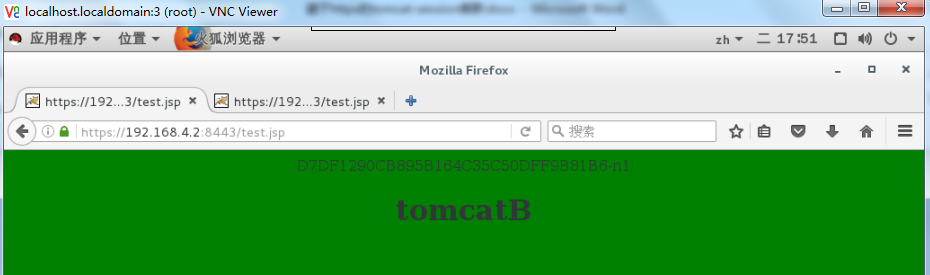
</body>

</html>

#创建session页面，进行测试



##另外一台tomcat做同样的设置，修改页面参数，颜色改为绿色，改成B



##tomcat部署完毕

##返回修改nginx

[root@localhost ~]# sed -n '98,120p' /usr/local/nginx/conf/nginx.conf

upstream tomcat {

server 192.168.4.3:8443;

server 192.168.4.2:8443;

}

server {

listen 443 ssl;

server\_name localhost;

ssl\_certificate cert.pem;

ssl\_certificate\_key cert.key;

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 5m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

location / {

proxy\_pass https://tomcat;

root html;

index index.html index.htm;

}

}

#设置后台集群代理

##安装memcached

[root@localhost ~]# yum -y install memcached

[root@localhost ~]# systemctl start memcached; systemctl enable memcached.service

Created symlink from /etc/systemd/system/multi-user.target.wants/memcached.service to /usr/lib/systemd/system/memcached.service.

[root@localhost ~]# netstat -anptu | grep memcached

tcp 0 0 0.0.0.0:11211 0.0.0.0:\* LISTEN 7492/memcached

tcp6 0 0 :::11211 :::\* LISTEN 7492/memcached

udp 0 0 0.0.0.0:11211 0.0.0.0:\* 7492/memcached

udp6 0 0 :::11211 :::\* 7492/memcached

##启动memcached确认监听端口

#修改tomcat配置文件

[root@localhost ~]# ls \*.jar | wc -l

9

[root@localhost ~]# mv \*.jar /usr/local/tomcat/lib/

[root@localhost ~]# cd /usr/local/tomcat/conf/

[root@localhost conf]# mv context.xml context.xml.bak

[root@localhost conf]# cp /root/context.default2 context.xml

[root@localhost conf]# vim context.xml

[root@localhost conf]# sed -n '35,42p' context.xml

<Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"

memcachedNodes="n1:192.168.4.1:11211"

sticky="false"

sessionBackupAsync="false"

lockingMode="none"

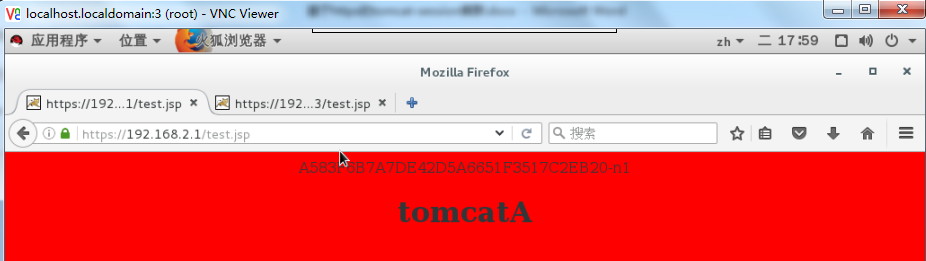
requestUriIgnorePattern=".\*\.(ico|png|gif|jpg|css|js)$"

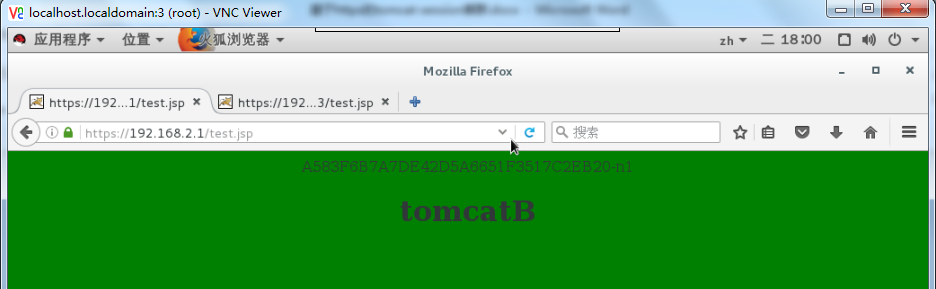
transcoderFactoryClass="de.javakaffee.web.msm.JavaSerializationTranscoderFactory"/>

</Context>

##两台同步操作

##测试，访问代理服务器的eth1





得到基于https协议的tomcat-session集群效果。